-continued

APPENDIX

```
                FALSE);
        IoSetCompletionRoutine (
                irp,
                synch_post,
                irp_context,
                TRUE,
                TRUE,
                TRUE);
        io_call_status =3D IoCallDriver (
                target_device,
                irp);
    KeWaitForSingleObject (
                &irp_context->event,
                Executive,
                KernelMode,
                FALSE,
                NULL);
    status =3D (*routine) (
                irp_context);
        ASSERT(status =3D=3D STATUS_SUCCESS);
        free_irp_context (
                irp_context);
            return io_call_status;
}
NTSTATUS synch_post (
                PDEVICE_OBJECT devobj,
                PIRP        irp,
                PIRP_CONTEXT irp_context)
{
        //
        // If IoCallDriver returned PENDING, mark our
// stack location with pending.
        //
    if (irp->PendingReturned)
    {
        IoMarkIrpPending( irp );
    }
    irp_context->io_status =3D irp->IoStatus.Status;
    irp_context->io_info =3D irp->IoStatus.Information;
    KeSetEvent (
                &irp_context->event,
                0,
                FALSE);
    return STATUS_SUCCESS;
}
```

While the invention has been described in terms of a single preferred embodiment, those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the appended claims.

Having thus described our invention, what we claim as new and desire to secure by Letters Patent is as follows:

1. A filter driver for use with an operating system, comprising:

means for establishing disk space quotas, said quotas being established in relation to a plurality of quota parameters;

means for determining disk space utilization in relation to each of said quota parameters;

means for storing said disk space quotas and said disk space utilization in a file structure;

means for monitoring disk I/O;

means for prospectively evaluating the effect of said disk I/O on said disk space utilization, said effect being a revised disk space utilization in relation to said disk space quotas;

means for terminating said disk I/O if said revised disk space utilization exceeds any of said disk space quotas; and

means for updating said file structure to reflect completion of said disk I/O if said revised disk space utilization does not exceed any of said disk space quotas.

2. The filter driver of claim 1, wherein said monitoring means further comprises:

means for detecting disk I/O operations, each said disk I/O operation being associated with a named disk file;

means for serializing operations on said named disk file.

3. The filter driver of claim 2, wherein said serializing means further comprises:

means for locking a synchronizing object, said synchronizing object being associated with said named disk file, and said locking means serving to block further disk I/O operations on said named disk file.

4. The filter driver of claim 3, wherein said synchronizing object is a kernel event in the Windows NT operating system.

5. The filter driver of claim 1, wherein said plurality of quota parameters comprise one or more ownership quotas and one or more directory quotas, each said ownership quota being a maximum quantity of said disk space in use by files associated with a particular owner, and each said directory quota being a maximum quantity of said disk space in use by files associated with a particular directory.

6. The filter driver of claim 1, wherein access to said file structure is serialized.

7. The filter driver of claim 1, wherein said monitoring means ignores paging I/O.

8. The filter driver of claim 1, wherein said prospective evaluation means further comprises:

means for determining said revised disk space utilization; and

means for comparing said revised disk space utilization to said disk space quotas.

* * * * *